Network Standard Data Specification Syntax

Network Working Group D. Crocker (UCLA-NMC)
Request for Comments: 645 ~~21 OCT 74~~ 26 JUN 74

NIC: 30899
Obsoletes: RFC # 615 (NIC # 21531)

Network Standard Data Specification Syntax

1

INTRODUCTION 2

This document defines the basic components of a Network Standard Data Specification (NSDS) syntax. A NSDS is intended to provide a mechanism for specifying all the attributes of a collection of bits. 2a

The definition of a complete NSDS syntax is expected to require an extended effort. Therefore the initial scope of this document has been constrained to provide only a basic syntactic environment. 2a1

In order to demonstrate a specific use for the NSDS, this document also provides the complete syntax for specifying the PATHNAME attributes of a collection of bits, to the level of a file. Addition of new subparameters should not be difficult. 2b

In this context, "pathname" refers to that information which specifies the LOCATION of a collection of bits. 2b1

The pathname syntax is essentially the same as that proposed in RFC 615 (NIC -- 21531,). Modifications were made in order to allow for graceful addition of other file attributes and to optimize use by humans and by processes. 2b2

I would like to thank Jon Postel, Jerry Popek, Vint Cerf, Jim White, Charlie Kline, Buz Owen, Ken Pogran, Jerry Burchfiel and Tom Boynton for their suggestions. 2c

HUMAN AND MACHINE FACTORS 3

Since computers tend to prefer more highly structured environments than do humans, aspects of the NSDS syntax are permitted to be different for computers than they are for humans. Specifically: 3a

For computers (highly-structured mode), keyword fields are fixed length and the variable-length data subfields are prefaced by a byte count. Additionally in highly structured mode, the possible contents of data subfields may be more constrained than for the semi-structured mode. 3a1

For humans (semi-structured mode), keyword subfields are variable length and data subfields are surrounded by delimeter characters. A keyword must be long enough to distinguish it from other keywords. That is, partial-name specification is permitted. 3a2

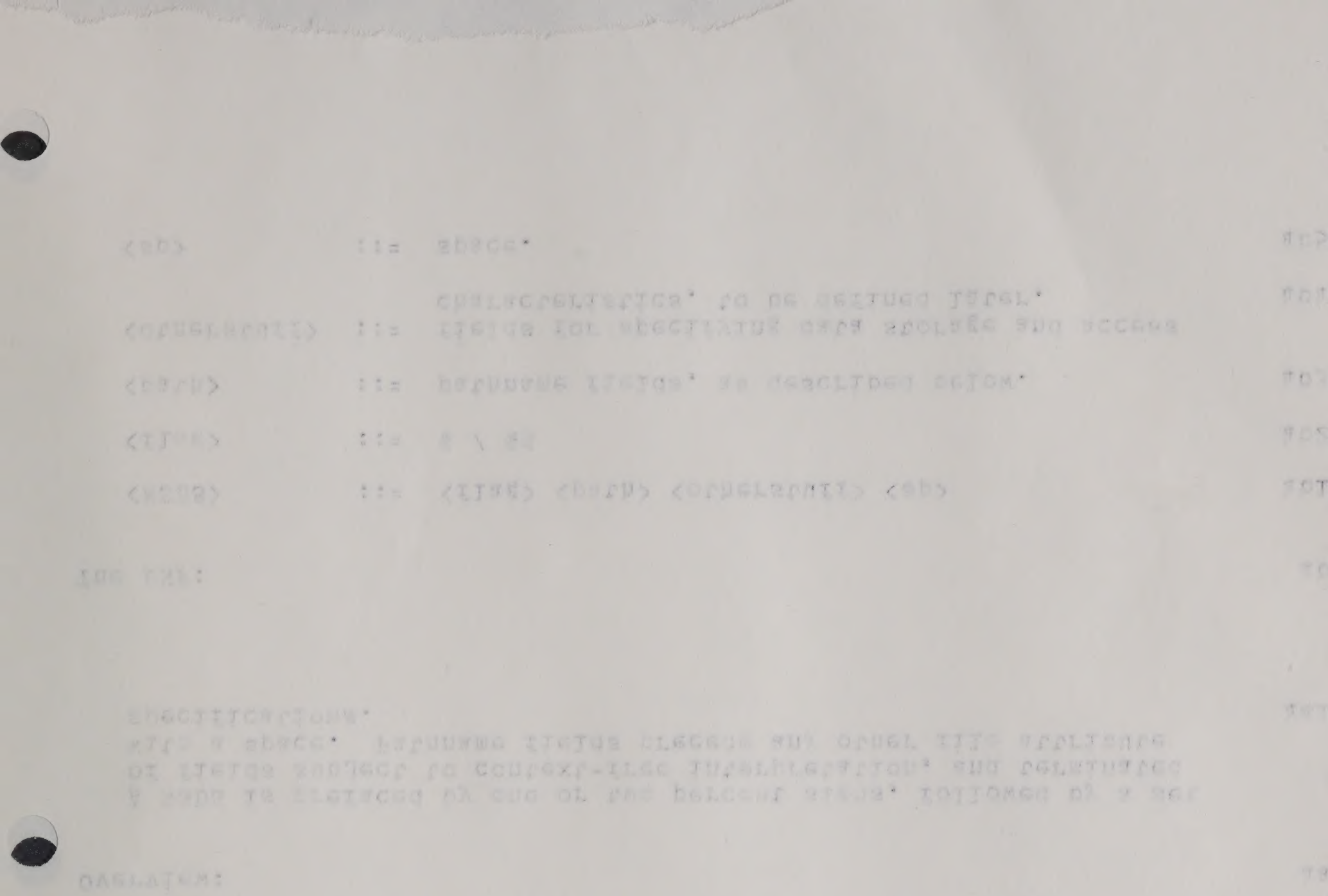
STRUCTURE OF THE GENERAL SYNTACTIC ENVIRONMENT 4

Overview: 4a

A NSDS is prefaced by one or two percent signs, followed by a set of fields subject to context-free interpretation, and terminated with a space. Pathname fields precede any other file attribute specifications. 4a1

The BNF: 4b

<NSDS>	::=	<flag> <path> <otherstuff> <sp>	4b1
<flag>	::=	% / %%	4b2
<path>	::=	pathname fields, as described below.	4b3
<otherstuff>	::=	fields for specifying data storage and access characteristics, to be defined later.	4b4
<sp>	::=	space.	4b5



Comments: 4c

The <flag> indicates escape-to-NSDS-syntax. One percent sign indicates semi-structured syntax, two indicate that highly-structured syntax is being used. 4c1

Only <flag> must be considered in relation to any host's current syntax. It is not currently known to conflict with any host's syntax. 4c1a

Exclamation mark (!) is the only other character that seems permissible (on the assumption that the character should be a graphic). Its use would cause minor problems at Multics; but more importantly as a graphic, it is too similar to the numeral "1". 4c1a1

The basic (highest-level) syntax for individual <path> and <otherstuff> fields is the same, as defined below. The remaining lower-level syntax (including permissible keywords and data subfield contents) for <otherstuff> fields is left for later. 4c2

BASIC UNITS OF SUBSTRUCTURE 5

Overview: 5a

A semi-structured field begins with a varying-length descriptor. The descriptor is followed by a varying-length data subfield, which is surrounded by delimeter characters. 5a1

Highly-structured fields have fixed-length descriptors, followed by a data byte-count, followed by the data. 5a2

BNF for individual fields: 5b

<field> ::= <machine> / <human> 5b1

<machine> ::= <stru-field> / <stru-field> <machine> 5b2

<stru-field> ::= <stru-key> <count> <data> 5b3

<stru-key> ::= 4-character field definition keyword; see below. 5b4

```
<alfa-key>      ::=  <alfa-literal> | <alfa-literal> <alfa-key>

<alfa-literal>  ::=  <alfa-key> <comma> <alfa>

<execute>       ::=  <alfa-literal> \ <alfa-literal> <execute>

<literal>       ::=  <execute> \ <number>
```

<count> ::= one-byte binary count of number of bytes of <data>. 5b5

<human> ::= <h-field> / <h-field> <human> 5b6

<h-field> ::= <h-key> <h-rest> 5b7

<h-key> ::= variable-length field definition keyword; see below. 5b8

<h-rest> ::= <l-delim> <data> <r-delim>
/ <l-delim> <data> <r-delim> <h-rest> 5b9

<l-delim> ::= any non-alphabetic printable character that is not in the succeeding <data> subfield and that is acceptable to the object site. For visual aesthetics and to facilitate human parsing, anytime <l-delim> is a left-bracket character (<, [, (, ¯), <r-delim> must be the complementary right-bracket character (>,],), |). 5b10

<r-delim> ::= either 1) the same character as <l-delim> or 2) if the <l-delim> character is a left-bracket character (<, [, (, ¯) then its complementary right-bracket (>,],), |). 5b11

<data> ::= any sequence of characters acceptable to the object site. This is the actual data subfield with the file, directory, device (or whatever) attribute value. 5b12

Elaboration: 5c

Case is irrelevant to the syntax, though some sites will care about case in <data> subfields. 5c1

The key (<stru-key> or <h-key>) indicates what part of the NSDS the next <data> subfield refers to. 5c2

<R-delim> and <l-delim> are used to delimit the beginning and end of the <data> subfield. 5c3

<Fields> for pathnames ARE order dependent, but defaulted ones may be omitted. The order is as indicated for <key>s, below. That is, Network, Host, ... Siteparm. 5c4

Keywords are used, even though pathname attributes are ordered, to facilitate the addition of new fields and to be consistent with the syntax for <otherstuff> fields which are expected to be unordered. 5c4a

<Field>s or <h-rest> subfields may be repeated, as permitted by the object site. A series of <h-rest> subfields, without any <h-key> subfields is interpreted as a series of <h-field>s with identical <key>s. 5c5

Also, note that since the syntax does not constrain the contents of <data> subfields, compound names within a single <data> subfield are allowed. The delimeter used to separate names within a <data> subfield must be different from <l-delim>/<r-delim> and the same as that used at the object site, since that is the only site which will be able to interpret the <data> subfield. 5c5a

The validity of any combiniation of <field>s is entirely site-dependent. For example, if a site will accept it, an NSDS with a Host field, and nothing more, may be permissible. 5c6

The validity of <data> subfields' contents is generally site-dependent. Some exceptions are noted below. 5c6a

PATHNAME ATTRIBUTES AND VALUES 6

The basic syntax does not need to be altered, to create the ability to specify pathnames. Only <key> values need to be defined. 6a

Definition of Pathname <key>s: 6b

The keyword for semi-structured mode is given first, followed by the keyword for highly-structured mode, if different. For highly-structured mode, keywords that are less than four characters should be padded with blanks at the right. 6b1

Semi	Highly	Meaning	
NETWORK	NET	Reference to the network (e.g., ARPA) connected to the HOST that contains or will contain the collection of bits.	6b3
HOST		Reference to host machine that contains or will contain the collection of bits. Also see section on "Numbers".	6b4
PERIPHERAL	PERI	Peripheral device being referred to.	6b5

VOLUME-ID	VOL	The volume (e.g., specific tape reel or disk pack) associated with the named peripheral device.	6b6
DIRECTORY	DIR	Name of directory which contains a pointer to the entity (directory or filename) specified in the following <field>.	6b7
FILE		Basic name of the file (data set).	6b8
TYPE		Optional modifier to filename. (Tenex calls it the Extension.)	6b9
VERSION	VER	Optional third part to basic filename. Usually used to distinguish updated files. The <data> subfield will usually contain a number.	6b10
SITEPARM	SITE	A parameter, such as an access specification or account number, peculiar to the object site. The contents of the <data> subfield must serve to identify what Siteparm is involved. Each site will be responsible for defining the syntax of Siteparm <data> subfields it will accept. Note that the SITEPARM field allows specification of other than pathname data (e.g., access and account number).	6b11

Some reserved PERIPHERAL <data>s: 6c

The alternate forms are merely for typing convenience and are not related to the semi/highly structure modes. 6c1

DISK or DSK:	Immediate, direct-access secondary storage.	6c2
ONLINE or ONL:	Whatever immediately-accessible (measured in fractions of a second) storage the user accesses by default; usually disk.	6c3
TAPE or TAP:	Industry-compatible magnetic tape.	6c4
TAPE7 or TP7:	7-Track industry compatible tape.	6c5
TAPE9 or TP9:	9-Track industry compatible tape.	6c6
DECTAPE or DEC:	DEC Tape.	6c7

OFFLINE or OFF:	Any tertiary storage; usually tape, though "devices" like the Datacomputer are permissible. The user should expect to wait minutes or hours before being able to access OFFLINE files.	6c8
LINE←PRINTER or LPT:	Any available line-printer.	6c9
DOCUMENT←PRINTER or DOC:	Upper/lower case line printer, preferably with 8 1/2" X 11" unlined paper.	6c10
PAPER←TAPE←READER or PTR:	Paper tape reader.	6c11
PAPER←TAPE←PUNCH or PTP:	Paper tape punch.	6c12
CARD←PUNCH or PUN:	Standard 80-column card punch.	6c13
CARD←READER or RDR:	Standard 80-column card reader.	6c14
OPERATOR or OPR:	System Operator's console.	6c15
CONSULTANT or CON:	On-line consultant.	6c16

DEFAULTS FOR PATHNAME <DATA> SUBFIELDS: 6d

Often, the appropriate default will be the last-used value. However, defaults will generally be context dependent. Consequently, the following defaults are offered only as guidelines: 6d1

Network:	ARPA.	6d2
Host:	The host interpreting the NSDS.	6d3
Peripheral:	ONLINE (DISK).	6d4
Volume←id:	Catalogued system space.	6d5
Directory:	The user's current "working" directory, usually set by the logon process.	6d6
Filename:	None.	6d7
Type:	None.	6d8
Siteparm:	None.	6d9

```
;
;
;
```

NUMBERS 7

The following scheme is recommended for specifying numbers in <h-field> data subfields: 7a

A sequence of numeric characters, optionally followed by a character indicating the radix. The default radix is ten. "H" indicates hexadecimal; "O" (oh) indicates octal; "B" indicates binary; and (gratuitously) "D" indicates decimal. 7a1

In <stru-field> data subfields, the number should be pure binary. Therefore, reference to a host on the Arpanet would require one 8-bit byte. 7b

GENERAL COMMENTS 8

The syntax is intended to be adequate for all hosts, so any given portion of it may be inappropriate for any given host. 8a

A site is expected to permit specifications in a given field iff that site already has a way of accepting the same information. 8a1

Having two modes of specification (highly- and semi-structured) may prove to be unnecessary. They are defined here merely as a convenience for experimentation. 8a2

I believe that modifications to the syntax will be graceful additions, rather than wholesale redesign, and thus can be deferred for a while. Currently, any undefined attributes must be specified in a Siteparm field. 8b

The first version of the syntax was a mix of Tenex and Multics conventions. That is: 8c

(Network)/Host/Peripheral:Directory>Filename.Type;Siteparm 8c1

Though visually more attractive and generally quicker to type, it lacks extensibility. For example, adding version number as a standard field would be difficult. 8d

It is asserted (conceded) that, as long as extensibility is kept as a design goal, no standardized /semi-structured/ syntax will be as pleasant to use as currently exists on some systems. 8e

SOME SAMPLE PATHNAMES 9

Pathnames in NSDS that occupy more than one line, below, do so only because they are too long for a single line. Bracketed numbers (e.g., <8>) indicate a single byte with the number as its decimal value. Blanks (spaces) are indicated by <sp>. 9a

My message file at ISI (<DCROCKER>MESSAGE.TXT;P770404): 9b

Semi-structured 9b1

%H[ISI]D<DCROCKER>F(MESSAGE>T(TXT)S/P770404/<sp> 9b1a

Highly-structured 9b2

%%HOST<1><86>DIR<sp><8>DCROCKERFILE<7>MESSAGETYPE<3>TXTSITE<7>P770404<sp> 9b2a

ARPO61.LAD.DOCUMENT at UCLA-CCN. (Note the use of multiple Directory fields): 9c

Semi-structured 9c1

%H[65]DIR[ARPO61][LAD]F[DOCUMENT]<sp> 9c1a

Highly-structured 9c2

%%HOST<1><65>DIR<sp><6>ARPO61DIR<sp><3>LADFILE<8>DOCUMENT<sp> 9c2a

>udd>CompNet>Map>Mail at Mit-Multics. (Note that the initial NSDS Directory <data> subfield is empty, in keeping with Multics' method of starting at the top of its directory structure): 9d

Semi-structured 9d1

%H(540)DI[]DI[udd][CompNet]D(Map)FIL(Mail)<sp> 9d1a

Highly-structured 9d2

%%HOST<1><44>DIR<sp><0>DIR<sp><3>uddDIR<sp><7>CompNetDIR<sp><3>MapFILE<4>Mail<sp> 9d2a

Network Standard Data Specification Syntax

Network Working Group
Request for Comments: 645

D. Crocker (UCLA-NMC)
~~21 OCT 74~~
26 JUN 74

NIC: 30899
Obsoletes: RFC # 615 (NIC # 21531)

Network Standard Data Specification Syntax

1

INTRODUCTION 2

This document defines the basic components of a Network Standard Data Specification (NSDS) syntax. A NSDS is intended to provide a mechanism for specifying all the attributes of a collection of bits. 2a

The definition of a complete NSDS syntax is expected to require an extended effort. Therefore the initial scope of this document has been constrained to provide only a basic syntactic environment. 2a1

In order to demonstrate a specific use for the NSDS, this document also provides the complete syntax for specifying the PATHNAME attributes of a collection of bits, to the level of a file. Addition of new subparameters should not be difficult. 2b

In this context, "pathname" refers to that information which specifies the LOCATION of a collection of bits. 2b1

The pathname syntax is essentially the same as that proposed in RFC 615 (NIC -- 21531,). Modifications were made in order to allow for graceful addition of other file attributes and to optimize use by humans and by processes. 2b2

I would like to thank Jon Postel, Jerry Popek, Vint Cerf, Jim White, Charlie Kline, Buz Owen, Ken Pogran, Jerry Burchfiel and Tom Boynton for their suggestions. 2c

HUMAN AND MACHINE FACTORS 3

Since computers tend to prefer more highly structured environments than do humans, aspects of the NSDS syntax are permitted to be different for computers than they are for humans. Specifically: 3a

For computers (highly-structured mode), keyword fields are fixed length and the variable-length data subfields are prefaced by a byte count. Additionally in highly structured mode, the possible contents of data subfields may be more constrained than for the semi-structured mode. 3a1

For humans (semi-structured mode), keyword subfields are variable length and data subfields are surrounded by delimeter characters. A keyword must be long enough to distinguish it from other keywords. That is, partial-name specification is permitted. 3a2

STRUCTURE OF THE GENERAL SYNTACTIC ENVIRONMENT 4

Overview: 4a

A NSDS is prefaced by one or two percent signs, followed by a set of fields subject to context-free interpretation, and terminated with a space. Pathname fields precede any other file attribute specifications. 4a1

The BNF: 4b

<NSDS>	::=	<flag> <path> <otherstuff> <sp>	4b1
<flag>	::=	% / %%	4b2
<path>	::=	pathname fields, as described below.	4b3
<otherstuff>	::=	fields for specifying data storage and access characteristics, to be defined later.	4b4
<sp>	::=	space.	4b5

Comments: 4c

The <flag> indicates escape-to-NSDS-syntax. One percent sign indicates semi-structured syntax, two indicate that highly-structured syntax is being used. 4c1

Only <flag> must be considered in relation to any host's current syntax. It is not currently known to conflict with any host's syntax. 4c1a

Exclamation mark (!) is the only other character that seems permissible (on the assumption that the character should be a graphic). Its use would cause minor problems at Multics; but more importantly as a graphic, it is too similar to the numeral "1". 4c1a1

The basic (highest-level) syntax for individual <path> and <otherstuff> fields is the same, as defined below. The remaining lower-level syntax (including permissible keywords and data subfield contents) for <otherstuff> fields is left for later. 4c2

BASIC UNITS OF SUBSTRUCTURE 5

Overview: 5a

A semi-structured field begins with a varying-length descriptor. The descriptor is followed by a varying-length data subfield, which is surrounded by delimeter characters. 5a1

Highly-structured fields have fixed-length descriptors, followed by a data byte-count, followed by the data. 5a2

BNF for individual fields: 5b

<field> ::= <machine> / <human> 5b1

<machine> ::= <stru-field> / <stru-field> <machine> 5b2

<stru-field> ::= <stru-key> <count> <data> 5b3

<stru-key> ::= 4-character field definition keyword; see below. 5b4

<count>	::=	one-byte binary count of number of bytes of <data>.	5b5
<human>	::=	<h-field> / <h-field> <human>	5b6
<h-field>	::=	<h-key> <h-rest>	5b7
<h-key>	::=	variable-length field definition keyword; see below.	5b8
<h-rest>	::=	<l-delim> <data> <r-delim> / <l-delim> <data> <r-delim> <h-rest>	5b9
<l-delim>	::=	any non-alphabetic printable character that is not in the succeeding <data> subfield and that is acceptable to the object site. For visual aesthetics and to facilitate human parsing, anytime <l-delim> is a left-bracket character (<, [, (, ¯), <r-delim> must be the complementary right-bracket character (>,],), |).	5b10
<r-delim>	::=	either 1) the same character as <l-delim> or 2) if the <l-delim> character is a left-bracket character (<, [, (, ¯) then its complementary right-bracket (>,],), |).	5b11
<data>	::=	any sequence of characters acceptable to the object site. This is the actual data subfield with the file, directory, device (or whatever) attribute value.	5b12

Elaboration: 5c

Case is irrelevant to the syntax, though some sites will care about case in <data> subfields. 5c1

The key (<stru-key> or <h-key>) indicates what part of the NSDS the next <data> subfield refers to. 5c2

<R-delim> and <l-delim> are used to delimit the beginning and end of the <data> subfield. 5c3

<Fields> for pathnames ARE order dependent, but defaulted ones may be omitted. The order is as indicated for <key>s, below. That is, Network, Host, ... Siteparm. 5c4

Keywords are used, even though pathname attributes are ordered, to facilitate the addition of new fields and to be consistent with the syntax for <otherstuff> fields which are expected to be unordered. 5c4a

<Field>s or <h-rest> subfields may be repeated, as permitted by the object site. A series of <h-rest> subfields, without any <h-key> subfields is interpreted as a series of <h-field>s with identical <key>s. 5c5

Also, note that since the syntax does not constrain the contents of <data> subfields, compound names within a single <data> subfield are allowed. The delimeter used to separate names within a <data> subfield must be different from <l-delim>/<r-delim> and the same as that used at the object site, since that is the only site which will be able to interpret the <data> subfield. 5c5a

The validity of any combiniation of <field>s is entirely site-dependent. For example, if a site will accept it, an NSDS with a Host field, and nothing more, may be permissible. 5c6

The validity of <data> subfields' contents is generally site-dependent. Some exceptions are noted below. 5c6a

PATHNAME ATTRIBUTES AND VALUES 6

The basic syntax does not need to be altered, to create the ability to specify pathnames. Only <key> values need to be defined. 6a

Definition of Pathname <key>s: 6b

The keyword for semi-structured mode is given first, followed by the keyword for highly-structured mode, if different. For highly-structured mode, keywords that are less than four characters should be padded with blanks at the right. 6b1

Semi	Highly	Meaning	
			6b2
NETWORK	NET	Reference to the network (e.g., ARPA) connected to the HOST that contains or will contain the collection of bits.	6b3
HOST		Reference to host machine that contains or will contain the collection of bits. Also see section on "Numbers".	6b4
PERIPHERAL	PERI	Peripheral device being referred to.	6b5

VOLUME-ID	VOL	The volume (e.g., specific tape reel or disk pack) associated with the named peripheral device.	6b6
DIRECTORY	DIR	Name of directory which contains a pointer to the entity (directory or filename) specified in the following <field>.	6b7
FILE		Basic name of the file (data set).	6b8
TYPE		Optional modifier to filename. (Tenex calls it the Extension.)	6b9
VERSION	VER	Optional third part to basic filename. Usually used to distinguish updated files. The <data> subfield will usually contain a number.	6b10
SITEPARM	SITE	A parameter, such as an access specification or account number, peculiar to the object site. The contents of the <data> subfield must serve to identify what Siteparm is involved. Each site will be responsible for defining the syntax of Siteparm <data> subfields it will accept. Note that the SITEPARM field allows specification of other than pathname data (e.g., access and account number).	6b11

Some reserved PERIPHERAL <data>s: 6c

The alternate forms are merely for typing convenience and are not related to the semi/highly structure modes. 6c1

DISK or DSK:	Immediate, direct-access secondary storage.	6c2
ONLINE or ONL:	Whatever immediately-accessible (measured in fractions of a second) storage the user accesses by default; usually disk.	6c3
TAPE or TAP:	Industry-compatible magnetic tape.	6c4
TAPE7 or TP7:	7-Track industry compatible tape.	6c5
TAPE9 or TP9:	9-Track industry compatible tape.	6c6
DECTAPE or DEC:	DEC Tape.	6c7

OFFLINE or OFF:	Any tertiary storage; usually tape, though "devices" like the Datacomputer are permissible. The user should expect to wait minutes or hours before being able to access OFFLINE files.	6c8
LINE←PRINTER or LPT:	Any available line-printer.	6c9
DOCUMENT←PRINTER or DOC:	Upper/lower case line printer, preferably with 8 1/2" X 11" unlined paper.	6c10
PAPER←TAPE←READER or PTR:	Paper tape reader.	6c11
PAPER←TAPE←PUNCH or PTP:	Paper tape punch.	6c12
CARD←PUNCH or PUN:	Standard 80-column card punch.	6c13
CARD←READER or RDR:	Standard 80-column card reader.	6c14
OPERATOR or OPR:	System Operator's console.	6c15
CONSULTANT or CON:	On-line consultant.	6c16

DEFAULTS FOR PATHNAME <DATA> SUBFIELDS: 6d

Often, the appropriate default will be the last-used value. However, defaults will generally be context dependent. Consequently, the following defaults are offered only as guidelines: 6d1

Network:	ARPA.	6d2
Host:	The host interpreting the NSDS.	6d3
Peripheral:	ONLINE (DISK).	6d4
Volume←id:	Catalogued system space.	6d5
Directory:	The user's current "working" directory, usually set by the logon process.	6d6
Filename:	None.	6d7
Type:	None.	6d8
Siteparm:	None.	6d9

NUMBERS 7

The following scheme is recommended for specifying numbers in <h-field> data subfields: 7a

A sequence of numeric characters, optionally followed by a character indicating the radix. The default radix is ten. "H" indicates hexadecimal; "O" (oh) indicates octal; "B" indicates binary; and (gratuitously) "D" indicates decimal. 7a1

In <stru-field> data subfields, the number should be pure binary. Therefore, reference to a host on the Arpanet would require one 8-bit byte. 7b

GENERAL COMMENTS 8

The syntax is intended to be adequate for all hosts, so any given portion of it may be inappropriate for any given host. 8a

A site is expected to permit specifications in a given field iff that site already has a way of accepting the same information. 8a1

Having two modes of specification (highly- and semi-structured) may prove to be unnecessary. They are defined here merely as a convenience for experimentation. 8a2

I believe that modifications to the syntax will be graceful additions, rather than wholesale redesign, and thus can be deferred for a while. Currently, any undefined attributes must be specified in a Siteparm field. 8b

The first version of the syntax was a mix of Tenex and Multics conventions. That is: 8c

(Network)/Host/Peripheral:Directory>Filename.Type;Siteparm 8c1

Though visually more attractive and generally quicker to type, it lacks extensibility. For example, adding version number as a standard field would be difficult. 8d

It is asserted (conceded) that, as long as extensibility is kept as a design goal, no standardized /semi-structured/ syntax will be as pleasant to use as currently exists on some systems. 8e

SOME SAMPLE PATHNAMES 9

Pathnames in NSDS that occupy more than one line, below, do so only because they are too long for a single line. Bracketed numbers (e.g., <8>) indicate a single byte with the number as its decimal value. Blanks (spaces) are indicated by <sp>. 9a

My message file at ISI (<DCROCKER>MESSAGE.TXT;P770404): 9b

Semi-structured 9b1

%H[ISI]D<DCROCKER>F(MESSAGE)T(TXT)S/P770404/<sp> 9b1a

Highly-structured 9b2

%%HOST<1><86>DIR<sp><8>DCROCKERFILE<7>MESSAGETYPE<3>TXTSITE<7>P770404<sp> 9b2a

ARPO61.LAD.DOCUMENT at UCLA-CCN. (Note the use of multiple Directory fields): 9c

Semi-structured 9c1

%H[65]DIR[ARPO61][LAD]F[DOCUMENT]<sp> 9c1a

Highly-structured 9c2

%%HOST<1><65>DIR<sp><6>ARPO61DIR<sp><3>LADFILE<8>DOCUMENT<sp> 9c2a

>udd>CompNet>Map>Mail at Mit-Multics. (Note that the initial NSDS Directory <data> subfield is empty, in keeping with Multics' method of starting at the top of its directory structure): 9d

Semi-structured 9d1

%H(540)DI[]DI[udd][CompNet]D(Map)FIL(Mail)<sp> 9d1a

Highly-structured 9d2

%%HOST<1><44>DIR<sp><0>DIR<sp><3>uddDIR<sp><7>CompNetDIR<sp><3>MapFILE<4>Mail<sp> 9d2a